Speech Recognition - Acoustic Modeling

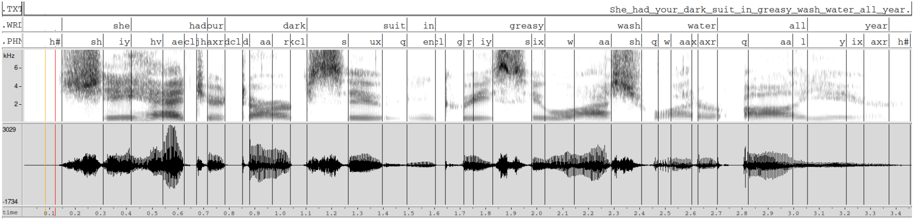
K Sri Rama Murty

IIT Hyderabad

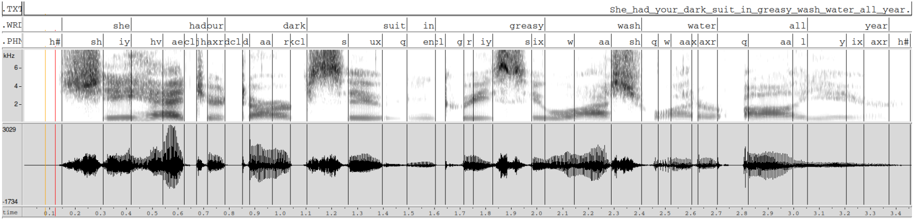
ksrm@ee.iith.ac.in

November 24, 2022

Speech Recognition

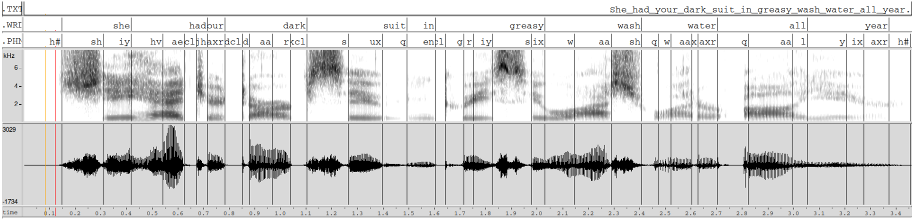


Speech Recognition



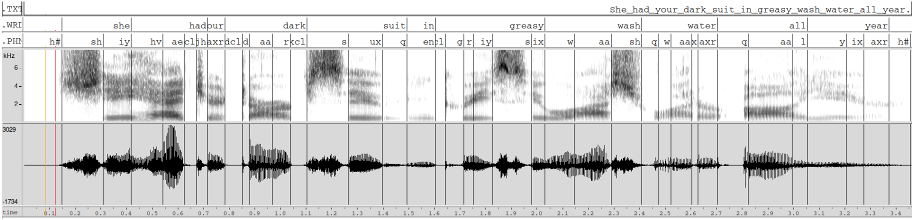
- The task of recognizing the text from the acoustic signal

Speech Recognition



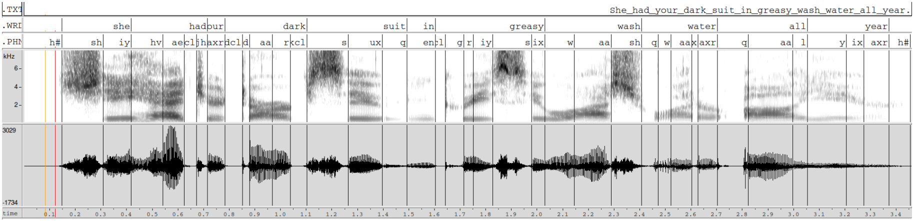
- The task of recognizing the text from the acoustic signal
- Attempted as a supervised learning task

Speech Recognition



- The task of recognizing the text from the acoustic signal
- Attempted as a supervised learning task
- Determine the best possible word sequence from the observed signal

Speech Recognition



- The task of recognizing the text from the acoustic signal
- Attempted as a supervised learning task
- Determine the best possible word sequence from the observed signal
- Time-domain samples → Feature representation → Subword units → Words → Sentences

Mathematical Formulation

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_{W} P[W/\mathbf{O}]$$

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_{W} P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \ldots w_k]$ is the probability of word sequence

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \ldots w_k]$ is the probability of word sequence
  - $p(\mathbf{O}/W)$ is the probability of observing feature $\mathbf{O}$ given W is uttered

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \dots w_k]$ is the probability of word sequence
  - $p(\mathbf{O}/W)$ is the probability of observing feature $\mathbf{O}$ given W is uttered
  - $p(\mathbf{O})$ is the partition function that normalizes the posterior estimates.

Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \ldots w_k]$ is the probability of word sequence
  - $p(\mathbf{O}/W)$ is the probability of observing feature $\mathbf{O}$ given W is uttered
  - $p(\mathbf{O})$ is the partition function that normalizes the posterior estimates.
- Most-likely word sequence can be determined by maximizing

$$W^* = \arg\max_W \underbrace{P[W]}_{LM} \; \underbrace{p(\mathbf{O}/W)}_{AM}$$

Acoustic Modeling

Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W

Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.

Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,

Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions

Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time

Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time
- $p(\mathbf{O}/W)$ is estimated using hidden Markov models (HMM)

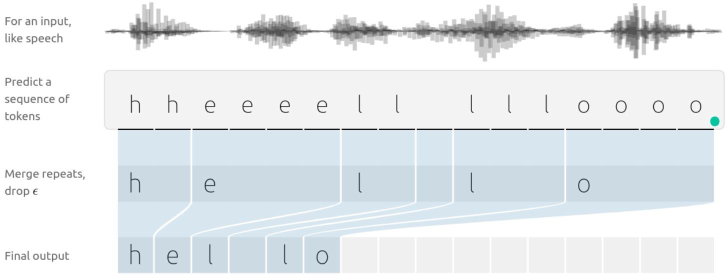
Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time
- $p(\mathbf{O}/W)$ is estimated using hidden Markov models (HMM)
- $P[W]$ is estimated using Markov models

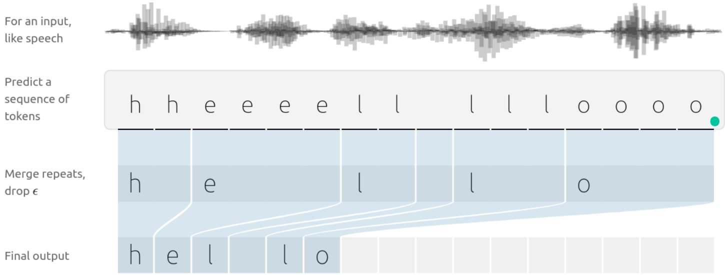
Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and W
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time
- $p(\mathbf{O}/W)$ is estimated using hidden Markov models (HMM)
- $P[W]$ is estimated using Markov models
- End-to-end neural network models directly estimate $P[W/\mathbf{O}]$

Towards End-to-End Speech Recognition



Towards End-to-End Speech Recognition



- Map acoustic observation sequence $O = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)$ to alphabet sequence $W = (w_1, w_2, \cdots w_U)$, where $W_k \in \{S_1, S_2, \cdots S_{26}\}$
  - The sequences O and W are of different length
  - The ratio of lengths of O and W can vary
  - Do not have access to accurate alignment between O and W

Sequence Model

- Train a model to infer word sequence W from observation sequence O
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

$$W^* = \arg\max_{\text{all } W} P[W/O]$$

Sequence Model

- Train a model to infer word sequence W from observation sequence O
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

$$W^* = \arg\max_{\text{all } W} P[W/O]$$

- How to proceed with assigning posteriors to word sequences?

Sequence Model

- Train a model to infer word sequence W from observation sequence O
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

$$W^* = \arg\max_{\text{all } W} P[W/O]$$

- How to proceed with assigning posteriors to word sequences?
    - For every o_t, assign a posterior distribution over all possible W

$$P[w_t = S_j/o_t] \qquad j = 1, 2, \cdots 26 \text{ and } t = 1, 2, \cdots T$$

    - Sequence of posteriors can be used to evaluate P[W/O]

Sequence Model

- Train a model to infer word sequence W from observation sequence O
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

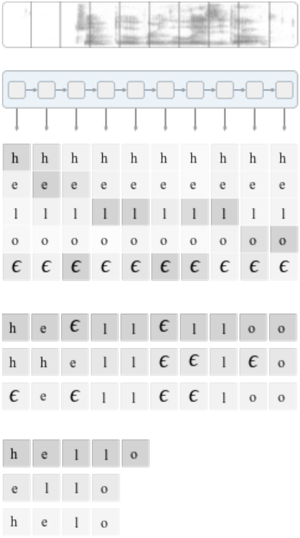
$$W^* = \arg\max_{\text{all } W} P[W/O]$$

- How to proceed with assigning posteriors to word sequences?
  - For every o_t, assign a posterior distribution over all possible W

$$P[w_t = S_j/o_t] \qquad j = 1, 2, \cdots 26 \text{ and } t = 1, 2, \cdots T$$

  - Sequence of posteriors can be used to evaluate P[W/O]
- RNNs/CNNs are used to map the observations to word posteriors
  - Cross-entropy loss cannot be used as it requires ground-truth alignment

Connectionist Temporal Classification (CTC))



We start with an input sequence, like a spectrogram of audio.

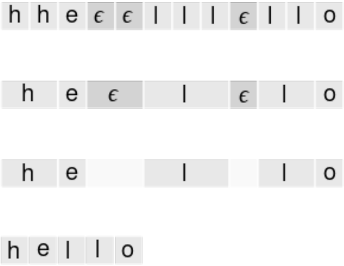
The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ } for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs.

CTC Alignment Steps

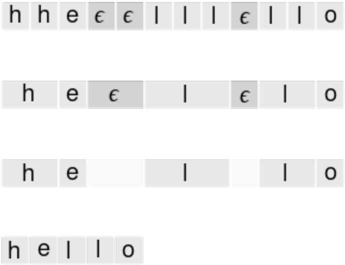


First, merge repeat characters.

Then, remove any ϵ tokens.

The remaining characters are the output.

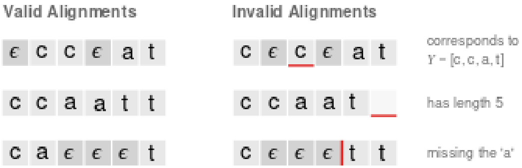
CTC Alignment Steps



First, merge repeat characters.

Then, remove any ϵ tokens.

The remaining characters are the output.



CTC Loss

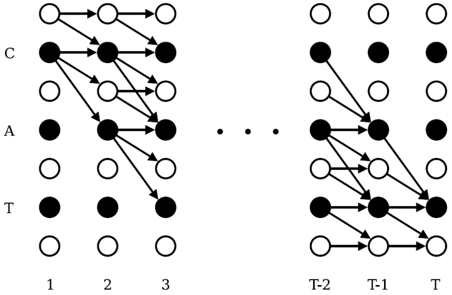
- Probability of a word sequence W given the observation sequence O

$$P[W/O] = \sum_{\text{all valid paths}} \prod_{t=1}^{T} P[w_t/\mathbf{o}_1, \mathbf{o}_2, \cdots \mathbf{o}_T]$$

- During training, manual transcription of words/sentences is known
  - Restrict output posterior computation to the alphabet in those words
  - Form the trellis by arranging posteriors in the order of alphabet
  - Evaluate the probabilities along all the paths resuting in the given word
  - Compute the gradients, and backpropagate to maximize the probability
- Negative logarithm of the $P[W/O]$ is referred to as CTC loss

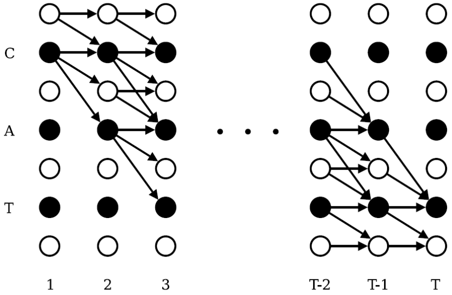
$$\mathcal{L}(\theta) = -\frac{1}{\mathcal{B}} \sum_{(O,W)\in\mathcal{B}} P[W/O]$$

CTC Forward Variable



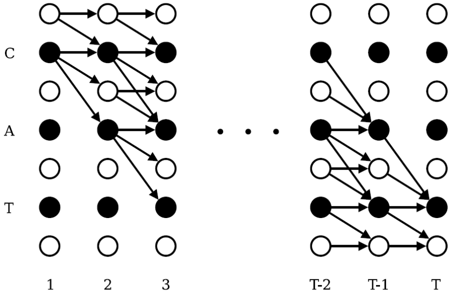
- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$

CTC Forward Variable



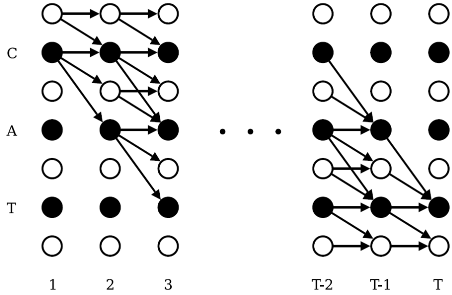
- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$
- $W = (w_1, w_2 \ldots w_L)$

CTC Forward Variable



- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$
- $W = (w_1, w_2 \ldots w_L)$
- $Z = (\epsilon, w_1, \epsilon, w_2, \ldots \epsilon, w_L, \epsilon)$

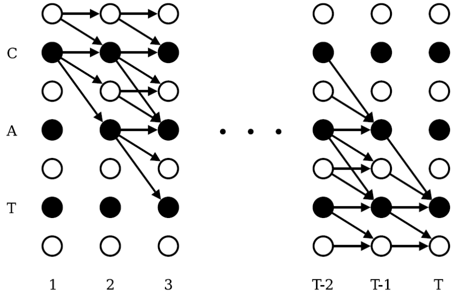
CTC Forward Variable



- Initialization

- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$
- $W = (w_1, w_2 \ldots w_L)$
- $Z = (\epsilon, w_1, \epsilon, w_2, \ldots \epsilon, w_L, \epsilon)$
- $\alpha_t(i) = p(a_1, a_2, \ldots, a_t = z_i / \mathbf{O})$

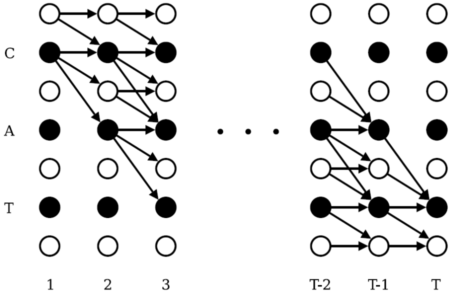
CTC Forward Variable



- Initialization
- Recursion

- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$
- $W = (w_1, w_2 \ldots w_L)$
- $Z = (\epsilon, w_1, \epsilon, w_2, \ldots \epsilon, w_L, \epsilon)$
- $\alpha_t(i) = p(a_1, a_2, \ldots, a_t = z_i / \mathbf{O})$

CTC Forward Variable

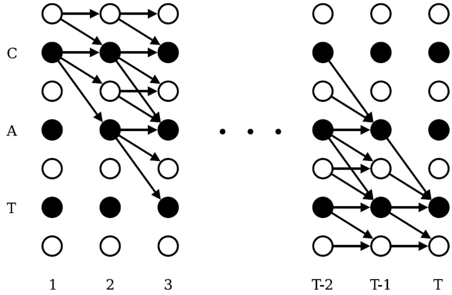


- Initialization
- Recursion
- Termination

$$P[W/\mathbf{O}] = \alpha_T(2L) + \alpha_T(2L+1)$$

- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$
- $W = (w_1, w_2 \ldots w_L)$
- $Z = (\epsilon, w_1, \epsilon, w_2, \ldots \epsilon, w_L, \epsilon)$
- $\alpha_t(i) = p(a_1, a_2, \ldots, a_t = z_i/\mathbf{O})$

CTC Forward Variable



- $\mathbf{O} = (\mathbf{o}_1, \ldots \mathbf{o}_t \ldots \mathbf{o}_T)$
- $A = (a_1, \ldots a_t \ldots a_T)$
- $W = (w_1, w_2 \ldots w_L)$
- $Z = (\epsilon, w_1, \epsilon, w_2, \ldots \epsilon, w_L, \epsilon)$
- $\alpha_t(i) = p(a_1, a_2, \ldots, a_t = z_i / \mathbf{O})$

- Initialization
- Recursion
- Termination

$$P[W/\mathbf{O}] = \alpha_T(2L) + \alpha_T(2L+1)$$

- Loss over a batch of examples

$$\mathcal{L}(\theta) = - \sum_{(O,W) \in \mathcal{B}} \log P[W/O]$$

- Loss function involves sums and products of posterior estimates of the network p(char/feature)

Network Training & Inference

- $P[W/O]$ can be efficiently computed using forward variable
- The procedure is similar to defining a forward variable for DTW

Network Training & Inference

- $P[W/O]$ can be efficiently computed using forward variable
- The procedure is similar to defining a forward variable for DTW
- During inference, evaluate posteriors over all the 26 alphabets
- Evaluate the best path over the trellis of size $(26 \times T)$
- Modified Viterbi algorithm can be used to arrive at the best path

$$W^* = \arg\max_W P[W/O]$$

Network Training & Inference

- $P[W/O]$ can be efficiently computed using forward variable
- The procedure is similar to defining a forward variable for DTW
- During inference, evaluate posteriors over all the 26 alphabets
- Evaluate the best path over the trellis of size $(26 \times T)$
- Modified Viterbi algorithm can be used to arrive at the best path

$$W^* = \arg\max_W P[W/O]$$

- What happend to the HMM transition probabilities?
- Issues:
  - Training should be done on shorter-utterances
  - Requires a huge amount of data (10k hours) for training

Inference

- Given λ, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

Inference

- Given λ, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

- Infer the likely word sequence for the given observation sequence

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

Inference

- Given λ, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

- Infer the likely word sequence for the given observation sequence

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- Consider most-likely output at each time-step

$$A* = \arg\max_A \prod_{t=1}^{T} p[a_t/\mathbf{O}]$$

Inference

- Given λ, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

- Infer the likely word sequence for the given observation sequence

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- Consider most-likely output at each time-step

$$A* = \arg\max_A \prod_{t=1}^{T} p[a_t/\mathbf{O}]$$

  - It results in alignment with highest probability
  - Collapse the repeats and remove ϵ to get W
  - Works well when most probability mass is allotted to a single alignment

Issue with Heuristic Approach

- Single output can have multiple alignments

$$a- > [a, a, a] || [a, a, \epsilon] || [\epsilon, a, a]$$

$$b- > [b, b, b]$$

- Probability along $[b]$ could be highest, however
- Sum of probabilities for $[a]$ could be higher than $[b]$

Issue with Heuristic Approach

- Single output can have multiple alignments

$$a- > [a, a, a] || [a, a, \epsilon] || [\epsilon, a, a]$$

$$b- > [b, b, b]$$

- Probability along $[b]$ could be highest, however
- Sum of probabilities for $[a]$ could be higher than $[b]$
- Need to consider multiple alignments resulting in same word!
- Need to modify regular beam search to account multiple alignments

Issue with Heuristic Approach

- Single output can have multiple alignments

$$a- > [a, a, a] || [a, a, \epsilon] || [\epsilon, a, a]$$

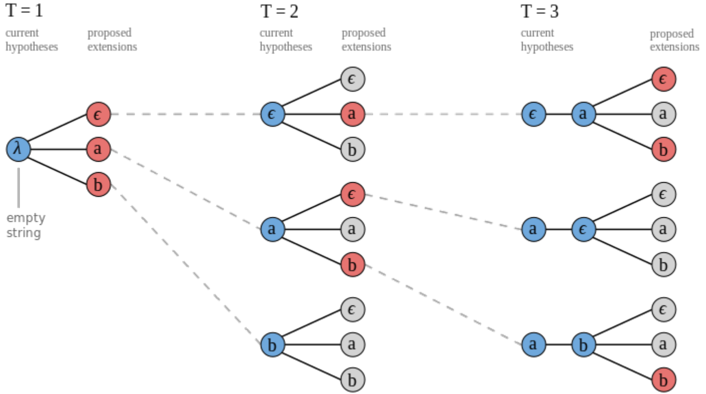
$$b- > [b, b, b]$$

- Probability along $[b]$ could be highest, however
- Sum of probabilities for $[a]$ could be higher than $[b]$
- Need to consider multiple alignments resulting in same word!
- Need to modify regular beam search to account multiple alignments
- Gamma variable used in beam search is similar to forward variable

$$\gamma_t(i) = \max_{a_1, a_2, \cdots a_{t-1}} p(a_1, a_2, \ldots, a_t = z_i / \mathbf{O})$$

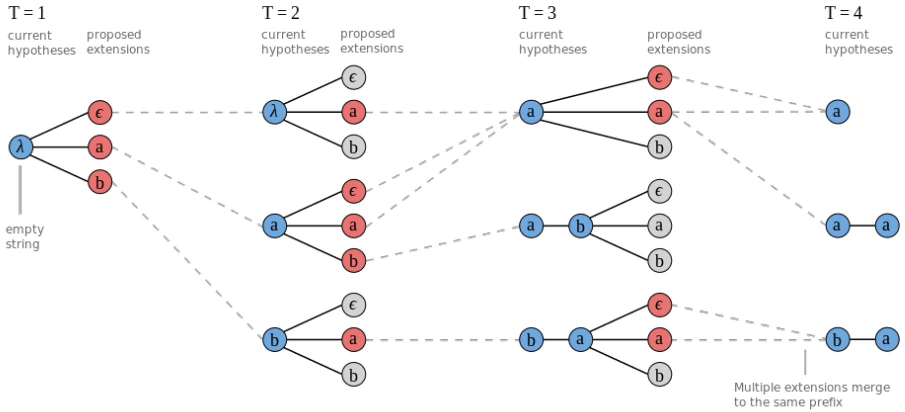
Beam Search

- Compute new set of hypothesis at each time-step
- Extend the previous best paths keeping only top candidates

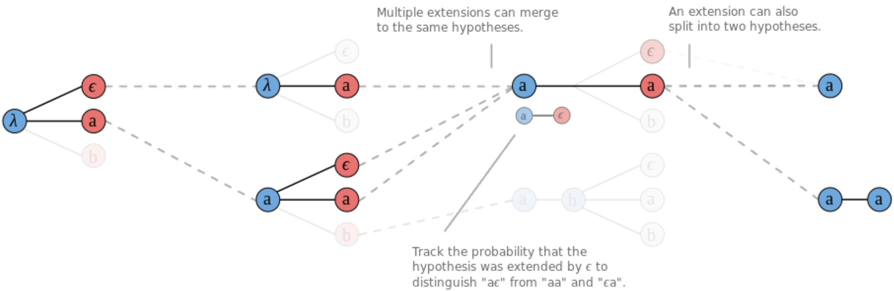


Modified Beam Search

- Instead of keeping list of alignments, store the output prefixes after collapsing repeats and removing ϵ characters

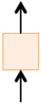


Tracking ϵ Character



RNN Configurations

Single - Single



Feed-forward Network

Single - Multiple

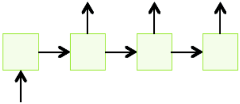
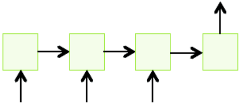


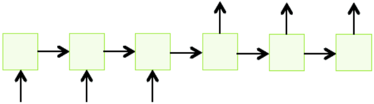
Image Captioning

Multiple - Single



Sentiment Classification

Multiple - Multiple



Translation

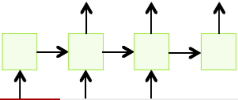
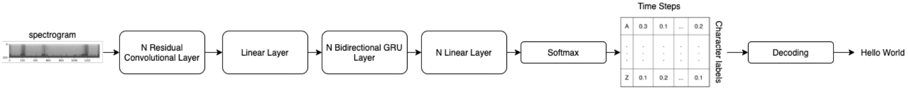
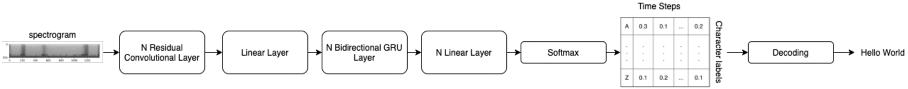


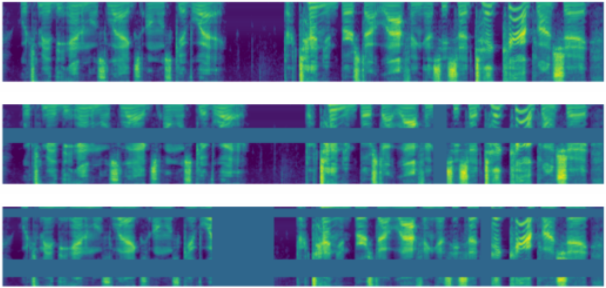
Image Captioning

End-to-End Speech Recognizer

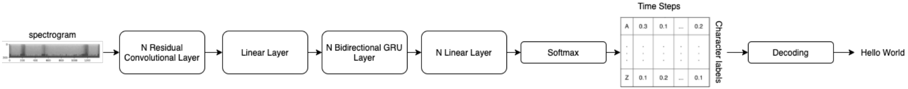


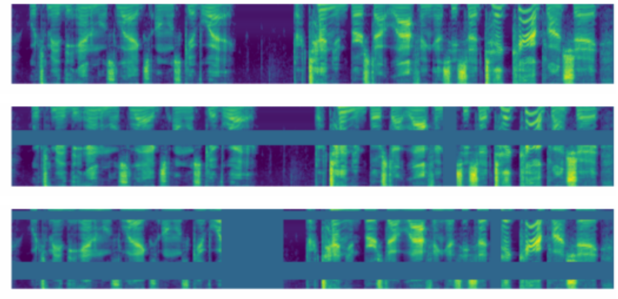
End-to-End Speech Recognizer





End-to-End Speech Recognizer





What is wrong with this spectrogram?

Source: https://www.assemblyai.com/blog/end-to-end-speech-recognition-pytorch/

Statistical Language Modeling

- Evaluate joint-probability of sequence of words occurring together

$$P[\mathbf{W}] = P[W_1, W_2, \cdots W_N]$$

- Condition upcoming word on the past history (chain rule)

$$P[\mathbf{W}] = P[W_1] \prod_{i=2}^{N} P[W_i / W_1, W_2, \cdots W_{i-1}]$$

- Not enough data to estimate probabilities over increasing contexts!
- Markov assumption - Restrict memory to fixed steps
- LM can be approximated under K^{th} order Markov assumption as

$$P[\mathbf{W}] = \prod_{i=1}^{N} P[W_i \ / \ W_{i-1}, \ W_{i-2}, \ W_{i-K}]$$

Effect of Model Order K

- $K = 0$: Unigram - words in a sentence are independent

$$P[\mathbf{W}] = \prod_{i=1}^{N} P[W_i]$$

  - young you wall last but and had in after n't words 'nothing more away
  - fifth an of futures the an incorporated a a the inflation most dollars

- $K = 1$: Bigram - Condition current word on the previous word

$$P[\mathbf{W}] = P[W_1] \prod_{i=2}^{N} P[W_i \ / \ W_{i-1}]$$

  - I must have taken into this way out of her by one hand
  - outside new car parking lot of the agreement reached

Estimating Bigram Probabilities

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

  where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together
- Example Data:

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

  where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together
- Example Data:
  - @ I am sam *

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together

- Example Data:
  - @ I am sam *
  - @ Sam I am *

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together

- Example Data:
  - @ I am sam *
  - @ Sam I am *
  - @ I do not like green eggs and ham *

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together

- Example Data:
  - @ I am sam *
  - @ Sam I am *
  - @ I do not like green eggs and ham *
- Estimated bigram probabilities

$$P(I/@) = 2/3 \quad P(Sam/@) = 1/3 \quad P(am/I) = 2/3$$

$$P(*/Sam) = 1/2 \quad P(Sam/am) = 1/2 \quad P(do/I) = 1/3$$

Estimating Bigram Probabilities

- Maximum likelihood estimate of Bigram probabilities is given by

$$P[W_i/W_j] = \frac{C(W_j, W_i)}{C(W_j)}$$

where $C(W_j, W_i)$ indicates the count of W_j, W_i occurring together

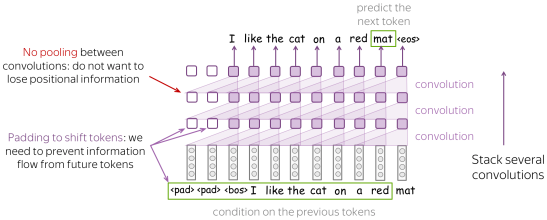
- Example Data:
  - @ I am sam *
  - @ Sam I am *
  - @ I do not like green eggs and ham *
- Estimated bigram probabilities

$$P(I/@) = 2/3 \quad P(Sam/@) = 1/3 \quad P(am/I) = 2/3$$

$$P(*/Sam) = 1/2 \quad P(Sam/am) = 1/2 \quad P(do/I) = 1/3$$

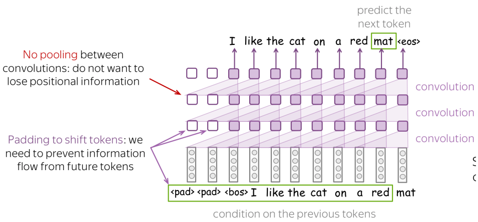
- Similar strategy can be employed to estimate n-gram LM

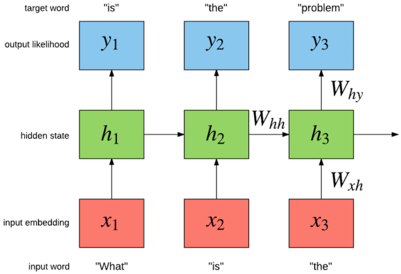
Neural Language Modeling



- CNN language model
- Offers finite context
- Easier to train

Neural Language Modeling





- CNN language model
- Offers finite context
- Easier to train

- RNN language model
- Infinite left context
- Capture long term dependencies

Feature Representation

- Speech production involves time-varying VTS and excitation
- For signal analysis, we assume stationarity over short intervals
  - Helps in applying concepts developed in LTI system theory
  - Easier to handle/relate time and frequency domain operations
- Short-term spectral analysis is commonly used for feature extraction
  - Features are extracted from 25ms frames shifted by 10 ms
    - Mel filter-bank energy coefficients, MFCCs, LPCCs
  - Each of the 25ms frame is analyzed in isolation
  - No explicit effort to capture relations among the sequence of frames
  - Hope to capture it implicitly in the frame overlap
- The burden of capturing sequence information is left to the "model"
- Can we incorporate sequence information into the features?

Representation Learning for Feature Extraction

- Importance of longer context in speech
  - Syntactic and semantic constraints of the language
    - Position dependent pronunciation of an alphabet
  - Learned behavioral characteristics of the speakers
  - Long-term prosodic patterns under different emotions
- Modeling high-level representations from raw observations
  - Should capture longer-contextual relations in the signal
  - Discard low-level information such as noise that is local
    - Isolated noise spurts, microphone differences, channel characteristics
    - Speaker-specific pronunciation differences for speech recognition

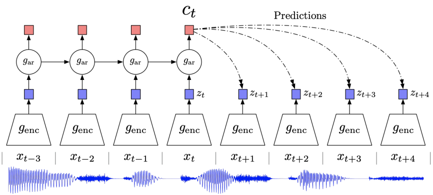
Predictive Coding

- Common strategy to capture context information is to predict future
  - Predictive coding has been used in signal processing & compression
  - Predictive coding was highly successful in language modeling
    - Predict next word based on the previous word
    - Predict the missing word from the neighboring words
- One way to predict x_t from the past is to enforce a generative model

$$x_t = g(x_1, x_2, \cdots, x_{t-1})$$

- Another is to impose autoregression on the underlying latent variable

Predictive Coding of Latent Information



- Let the observed speech frame be generated from latent variable z

$$z_t = g_{enc}(x_t)$$

- Let latent variables follow an AR process with encapsulated history

$$c_t = g_{ar}(z_1, z_2, \cdots z_t) = g_{ar}(z_{\leq t})$$

- Let future of z be predicted from latent contextual information c_t

$$z^p_{t+k} = W_k c_t$$

Model Parameter Estimation

- Both prediction z_{t+k} and context $z_{\leq t}$ depend on the same c_t
- We have two estimates for z_{t+k}: measurement & prediction

$$z^m_{t+k} = g_{enc}(x_{t+k}) \qquad z^p(t+k) = W_k c_t$$

- Estimate the model parameters (g_{enc}, g_{ar}, W_k) to improve coherence between the two estimates
- Maximize the mutual information between x_{t+k} and c_t

$$I(x,c) = \sum_{x,c} p(x,c) \log \frac{p(x,c)}{p(x)p(c)} = \sum_{x,c} p(x,c) \log \frac{p(x/c)}{p(x)}$$

- M.I estimation depends on the density ratio $f(x_{t+k,c_t}) = \frac{p(x_{t+k}/c)}{p(x)}$

Noise Contrastive Loss

- Let the ratio $f(x_{t+k,c_t})$ be evaluated as

$$f_k(x_{t+k,c_t}) = \exp(z^m_{t+k} W_k c_t)$$

- Maximizing MI is equivalent to minimizing NCE

$$\mathcal{L} = -\mathbb{E}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j} f_k(x_j, c_t)}\right]$$

- Numerator is computed using future frames, and denominator is computed using random frames
- The nature of the features depend on the choice of denominator
- Around 60k hours of unlabeled speech data is used to train the model
- After training, the latent variables z_t are used as features

Thank You!